Developing quality heathcare software using quality function deployment: A case study based on Sultan Qaboos University Hospital

Ahmed, Mohiuddin; Islam, Rafikul and Alwahaibi, Salim Khloof

International Islamic University Malaysia (IIUM)

2006

Developing quality healthcare software using quality function deployment: a case study based on Sultan Qaboos University Hospital

Mohiuddin Ahmed

Department of Computer Science
Faculty of Information and Communication Technology
International Islamic University Malaysia
Jalan Gombak, 53100 Kuala Lumpur, Malaysia
E-mail: mohiudin@iiu.edu.my

Rafikul Islam*

Department of Business Administration
Faculty of Economics and Management Sciences
International Islamic University Malaysia
Jalan Gombak, 53100 Kuala Lumpur, Malaysia
E-mail: rislam@iiu.edu.my
*Corresponding author

Salim Khloof Al-wahaibi

Sultan Qaboos University Hospital
P.O. Box 38, Alkhod 123, Sultanate of Oman
E-mail: swahaibi@sqh.edu.om

Abstract: Traditionally, the rate of failure in software development projects is higher compared to other kinds of projects. This is partly due to the failure in determining software users' requirements. By using Quality Function Deployment (QFD), this research focuses on identification and prioritisation of users' requirements in the context of developing quality healthcare software system for Sultan Qaboos University Hospital (SQUH) in Oman. A total of 95 staff working at eight departments of SQUH were contacted and they were requested to provide their requirements in using hospital information systems. Analytic Hierarchy Process has been integrated with QFD for prioritising those user requirements. Then, in consultation with a number of software engineers, a list consisting of 30 technical requirements was generated. At the end of QFD exercise the technical requirements that receive higher weights, should be paid due consideration at the time of designing the healthcare software system for SQUH.

Keywords: software quality; quality function deployment; healthcare software; analytic hierarchy process.

Reference to this paper should be made as follows: Ahmed, M., Islam, R. and Al-wahaibi, S.K. (2006) 'Developing quality healthcare software using quality function deployment: a case study based on Sultan Qaboos University Hospital', *Int. J. Business Information Systems*, Vol. 1, No. 4, pp.408–425.

Biographical notes: Mohiuddin Ahmed is an Assistant Professor in the Computer Science Department of the International Islamic University Malaysia (IIUM). Before joining IIUM, he was working in Goldman Sachs, NY, USA, as a Programmer Analyst in the Global Control Division. Prior to that, he was an adjunct faculty in the Department of Electrical and Computer Engineering in North Carolina A&T State University (NC A&T SU). He obtained his MS and PhD both in Electrical Engineering from NC A&T SU, USA in 1993 and 2000, respectively. His current research interests include High Performance Computing (HPC), Artificial Neural Network (ANN), ANN application in signal and image processing, *etc.*

Rafikul Islam received his MSc degree in Applied Mathematics from Calcutta University in 1988. Subsequently, he obtained his PhD in Operations Research from the Indian Institute of Technology, Kharagpur in 1996. Presently, he is working as an Associate Professor at the Department of Business Administration, International Islamic University Malaysia. His articles have appeared in *European Journal of Operational Research*, *Military Operations Research*, *International Journal of Fuzzy Mathematics*, *Ricerca Operativa*, *Journal of Engineering Manufacture*, *etc.* Dr. Islam's research area includes multiple criteria decision making, operations and quality management.

Salim Khloof Al-Wahaibi received his Bachelor's degree from New Hampshire College, London in 1988, and his MBA in Management Information System from the International Islamic University Malaysia in 2004. He was one of the founding members of the Sultan Qaboos University Hospital. In 1989, he was appointed as Medical Records Officer, and he started the Medical Records Department from scratch before the opening of the Hospital. He is currently working as Head of the Medical Records Department, Sultan Qaboos University Hospital, Oman.

1 Introduction

Software quality has become a topic of increasing importance during the past decade. The quality of software is primarily determined by the quality of the software development process. Pai (2002) mentions that even a small improvement in the software development process can result in a significant improvement of software quality. The software development process in the management of information systems has a history of being plagued by many problems. Most of these problems are associated with the specification of user requirements, which, when incorrect, incomplete or inconsistent, lead to significant budget overruns through increased programming and testing costs and product reworks (Hagg *et al.*, 1996).

Quality, unfortunately, is an ambiguous concept, is hard to define and often difficult to measure. This is especially true for Information Technology (IT) related products. Although there are many different methods that exist to control, measure, manage and improve quality in various other areas, little attention has been paid to measure quality of IT related systems (Tan *et al.*, 1998).

In the context of tangible products, Berk and Berk (2000) have simplified the problem of measuring quality by addressing the questions: How much of the product has to be scrapped because it does not meet dimensional or other requirements, and what is the

corresponding cost? How much of the product has to be reworked or repaired either during the manufacturing process or after delivery to the customer and how much is the cost? What are the largest areas of scrap, rework, or repair? Which of the above problems should we be fixing or working on first? In this connection, it is assumed that quality, which is usually based on a product's compliance to expectations and requirements, is a measurable characteristic and that measurement is based on the quantity and costs of non-conformance. It is also contended that poor quality raises costs tremendously, as poor quality increases the size and cost of the hidden factors (*e.g.*, scrap and rework). Value and quality of products can be improved by measuring non-conformance cost and systematically removing the dominant factors that cause non-conformance.

Software quality has traditionally been defined in terms of fitness for use (Ross *et al.*, 1995). A software product is deemed fit for use if it performs to some extent of user satisfaction in terms of functionality and continuous operation. For developing high quality software, an important part of the development process is the choice of the programming language (Berk and Berk, 2000). According to the authors, there are mainly three aspects that have to be taken in consideration regarding a programming language and its usage which has direct implications in software quality. These aspects are: the design of the programming language, the specification of the programming language, and its incorporation into a programming language standard.

Fitzpatrick (2001) presents 11 issues (strategic drivers) beyond software life-cycle process, which build a new conceptual model called Software Quality-Strategic Drivers Model (SQ-SDM). He defines Strategic Quality Driver as a set of interrelated issues, which must be managed (planned, organised, controlled and directed) in order to achieve success in a specific domain in a specified context. The various forms of Strategic Quality Driver that have impact on the procurer are technical excellence (supportability), user acceptance (acceptability), corporate alignment (alignability), statutory conformance (conformability), investment efficiency (affordability) and competitive support (superiority). The Strategic Quality Drivers that impact the producer are competitive excellence (domination), corporate accreditation (certification), domain specialty (qualification), development excellence (organisation) and quality management (direction). He concludes that in addition to its importance to software suppliers and acquirers, the SQ-SDM is an excellent foundation for the academic syllabus for the study of software quality.

Quality Function Deployment (QFD) is an excellent tool to design a product in response to customers' needs (Terninko, 1997; Silva, 2004; Ertay *et al.*, 2005). Since its development by Professor Mizuno in 1972, the tool has been used in numerous areas (Motwani and Kathawala, 1994; Franceschini and Marco, 1998; Ermer and Kniper, 1998; Hamza *et al.*, 1999). QFD has been used extensively to design various services using internet. Lin *et al.* (2005) have evaluated the performance of the after-sales service information systems provided by the Taiwanese machine tool industry. The authors conclude that web-based information management system must be used properly, and it can provide an important tool for interaction between enterprises and clients as well as for strengthening corporate operations and competitiveness. Using QFD as the link to customer requirements, González *et al.* (2004) have analysed various options used in the introduction of e-banking in the National Bank of Spain. The authors conclude that project managers and quality improvement managers could benefit from the QFD methodology in linking customer requirements to the internal procedures of the firm in order to satisfy customer requirements.

Grenci (2004–2005) has developed an online customer decision support system that can offer product configuration functionality. The author claims that this is a useful prototype that can provide a basis for organising and advancing important implications with respect to the key features and enablers of online customer configuration. Hamilton and Selen (2004) have discussed how integrated services can be delivered over the web in a service chain involving multiple parties. The researchers have used QFD to develop a framework that can fulfil user needs by suggesting appropriate service delivery in terms of web interface and content.

Barnett and Raja (1995) were the early researchers who proposed the use of QFD in developing software. Their proposed Software Quality Function Deployment (SQFD) model supports the activities in developing software for organisations. Tan *et al.* (1998) studied designing IT-related products using QFD. They focused at IT service as its importance was growing to the society and it was useful for the designers to provide services that best meet user needs. They also focused on how to improve an existing system as the life-cycle time for an IT product was usually very short. The findings of the authors' study showed that QFD method was feasible for defining, measuring and improving IT-related systems.

Hagg *et al.* (1996) presented the results of a survey on the adaptation and use of QFD for software development by major software companies. Thirty-seven companies were included in the study. Eighty percent of the organisations stated that the project leaders seriously considered the use of QFD in developing software. The leading purposes were analysing user demands, setting breakthrough targets and measuring the performances of competitors. SQFD was considered to be a step-by-step process for user satisfaction. Lack of management directive was found to be the main barrier and 33% cited no disadvantages concerning the use of SQFD. It was found that SQFD had a significant positive impact on user involvement, system development life cycle and project development.

The following section describes the research methodology used in conducting the present study to implement QFD tool in developing quality healthcare software for the healthcare institutes in general and the Sultan Qaboos University Hospital (SQUH) in particular.

2 Research methodology

The present research pertains to the 500 bed hospital of Sultan Qaboos University. The hospital has all the major departments and more than 200 000 patients' records are stored in its information systems databases. In a hospital, staff belonging to every department uses some kind of information system. Therefore, it is important to know all kinds of users' requirements in using the information systems. The respondents for this study comprise 95 staff from all the eight departments of the hospital. Data were collected from doctors, nurses, staff in-charge of operation theatres, *etc.*

Focus group is one of the most common approaches of data collection. The advantage of focus group for data collection is the face-to-face interaction between the data collectors and the respondents. In view of this, in the first phase of the study, several focus groups were established with the 95 staff of SQUH. There were eight groups, each representing a department. Each group had a group leader and the third author of this

paper was the coordinator for all the group leaders. Each group leader met with his/her group members and brainstormed to produce a list of factors that were important to them. Specifically, the group members were asked to articulate their requirements or expectations in using software systems in their respective departments. After this brainstorming exercise, the coordinator and all the group leaders assembled and synthesised all the lists by forming a master list (deleting the repeated ones). This information is valuable in the sense that opinions and expectations were captured in the users' words, not translated or filtered by some technical person. The final list of user requirements has been prioritised to find out the importance level using the Analytic Hierarchy Process (AHP). The reasons for using AHP to prioritise the list are its simplicity and application of the technique increases exactness in the priorities of the items in the list.

In the second phase, we have conducted several interviews with information system specialists and software engineers in order to find out engineering characteristics or technical requirements that will affect one or more user requirements. The interviewees were selected from the Faculty of Information and Communication Technology of International Islamic University Malaysia (IIUM) and the Information System Department of Sultan Qaboos University Hospital. Software engineers and information system specialists are in a better position to specify the engineering characteristics that can satisfy the users' articulated needs.

We provide brief descriptions of the Analytic Hierarchy Process and Quality Function Deployment in the following two sections.

3 The analytic hierarchy process

The Analytic Hierarchy Process (AHP) (Saaty, 1977; Islam, 2003) is a technique to derive ranking of a finite number of alternatives based upon a finite number of objectives (or criteria). To derive ranking with respect to some specific objective, all the alternatives are compared in a pairwise fashion. The typical form of a Pairwise Comparison Matrix (PCM) is as follows:

$\mathbf{A} =$

Obj. 'O'	A_1	A_2	…	A_n
A_1	a_{11}	a_{12}	…	a_{1n}
A_2	a_{21}	a_{22}	…	a_{2n}
…	…	…	…	…
A_n	a_{n1}	a_{n2}	…	a_{nn}

where $a_{ij} = \frac{w_i}{w_j}$ (for $i, j = 1,2,\ldots,n$) represents the strength of preference of the alternative A_i over A_j with respect to the objective 'O', $a_{ji} = \frac{1}{a_{ij}}$ and $a_{ii} = 1$ for all i, j. w_i, $i = 1,2,\ldots,n$ are the priority weights (to be determined) of the alternatives. The entries a_{ij} s are normally taken from the 1–9 ratio-scale (Saaty, 1977). The semantic interpretation of the numbers is provided in Table 1.

Table 1 Semantic interpretation of the ratios in the comparison matrices

Verbal judgment of preference	*Numerical rating*
Equally preferred	1
Equally to moderately preferred	2
Moderately preferred	3
Moderately to strongly preferred	4
Strongly preferred	5
Strongly to very strongly preferred	6
Very strongly preferred	7
Very strongly to extremely preferred	8
Extremely preferred	9

Note: If alternative A_i has preference strength as any of the above non-zero numbers compared to A_j, then A_j has the reciprocal value when compared with A_i, *i.e.*, $a_{ji} = 1/a_{ji}$.

The priority weights of all the alternatives can be derived by using the following simple geometric mean formula (Crawford, 1987):

$$w_i = \left(\prod_{j=1}^{n} a_{ij} \right)^{1/n}, i = 1, 2, ..., n. \tag{1}$$

However, to extract the weights from a pairwise comparison matrix, mathematically more rigorous method is to find out the largest eigenvalue of Matrix A and then compute the corresponding normalised eigenvector. The components of this normalised eigenvector give the weights of the alternatives. In practice, it was shown that the weights obtained by the above geometric mean rule and eigenvector method are quite close to each another. In this paper, to find out the weights, we have used Expert Choice software which uses eigenvector method.

4 Quality Function Deployment (QFD)

QFD is a graphical analysis technique that portrays customer (in the present paper, customers are the users of information systems) needs and expectations, how these needs and expectations are satisfied, and makes trade-off between conflicting needs and expectations (Bossert, 1991; Sanford, 2005). The analysis uses a diagram known as House of Quality (HOQ). At the first step of using HOQ diagram, customer requirements are determined for a product (in the present paper information system) and then a number of design characteristics (called technical requirements) are identified that are linked to the customer requirements. The technical requirements are expected to satisfy the customer requirements. In QFD terminologies, customer requirements and technical requirements are respectively known as 'WHATs' and 'HOWs'. After identifying the customer and technical requirements, the next step is to find out the relationships between these two types of requirements. It is to be noted that one single customer requirement may be related to more than one technical requirement, or vice versa. The matrix showing the relationship between customer and technical requirements is used to represent

graphically the degree of influence between each technical requirement and each customer requirement. Usually, three symbols are used to represent the relationships. These are shown below:

Symbol	Meaning	Weight
●	Strong relationship	9
○	Medium relationship	3
△	Weak relationship	1

If any technical requirement is not related, then the corresponding cell in the grid matrix is kept blank. Once we have identified all possible relationships between every pair of customer and technical requirements, we can determine the weights of all the technical requirements. The formula to compute the weights is:

$$t_j = \sum_{i=1}^{n} c_i r_{ij}, \qquad j = 1, 2, ..., n \tag{2}$$

where:

t_j = weight of the *j*-th technical requirement

c_i = weight of the *i*-th customer requirement (in the present study this is determined by using AHP)

r_{ij} = weight of the relationship between *i*-th customer requirement and *j*-th technical requirements

5 Data collection

Presently, all kinds of people working in a hospital use information systems to do their work. Consequently, they have their requirements as the users of the systems. As mentioned before, 95 staff from all the eight departments participated in the present study. Department wise break-up of the participants is shown in Table 2.

Participants from each department formed a focus group and generated a list of needs and requirements in response to the question: "what do you wish to have in the new hospital information system?" The lists are shown in the Appendix.

From the lists of user requirements shown in the Appendix, we created a synthesised list of user requirements. This synthesised list has been obtained by considering the common and seemingly more important requirements belonging to various individual (or departmental) lists. The synthesised list is shown in Table 3.

Having obtained the list of user requirements, we need to prioritise them according to their level of importance from the users' point of view, as they (the requirements) are not equally important. One participant from each group was requested to prioritise the requirements using (1–9) scale of AHP (described before). After collecting all individual pair-wise comparison matrices that have been filled up, we combined them and created one synthesised PCM for importance of user requirements using geometric mean rule of AHP (Basak and Saaty, 1993). Table 4 shows the synthesised pair-wise comparison matrix.

Table 2 Department wise participants

Number	*Department*	*Number of participants*
1	Anesthetic	4
2	Doctors	8
3	Nursing	14
4	Radiology	7
5	Medical records	4
6	Laboratories (microbiology, bio-chemistry, hematology, histopathology)	31
7	Pharmacy	5
8	Administration and finance	22
	(Administration, finance, purchasing, stores, maintenance, domestic services, training and staff development, bio-medical, housing)	
Total		*95*

Table 3 Synthesised user requirements

Code No.	*Requirements*
J1	Very fast
J2	User friendly
J3	One and unique Medical Record Number (MRN) should be capable of giving the details of a particular patient
J4	Interface with other instruments
J5	Automated lab reports and possibility to print all types of reports
J6	Departments stock check and update
J7	Incorporated internet facility
J8	Daily, monthly and yearly statistics
J9	Should keep daily, monthly and yearly backup
J10	Security and confidentiality
J11	Scanning related patients' paper records and integrating them with Electronic Medical Record (EMR)
J12	Capability for an advanced 'Alert Mail' service
J13	Very strong query system across all the modules
J14	Should have facility to support financial information
J15	Bi-lingual capability (English/Local language) is required in specific areas such as patients' registration data and instruction to patients in the medication and investigation areas

Table 4 Synthesised pairwise comparison matrix on user requirements

	J1	*J2*	*J3*	*J4*	*J5*	*J6*	*J7*	*J8*	*J9*	*J10*	*J11*	*J12*	*J13*	*J14*	*J15*	*Wts*
J1	1.000	1.260	0.475	0.737	0.511	2.621	3.107	0.385	0.291	0.303	1.101	1.063	0.322	3.271	0.606	0.070
J2		1.000	0.511	0.874	0.794	2.714	2.759	1.063	0.317	0.329	2.289	3.271	0.368	4.932	1.710	0.062
J3			1.000	2.520	1.145	2.759	3.107	0.550	0.500	0.481	3.107	3.826	1.000	2.520	2.924	0.087
J4				1.000	2.154	2.466	2.520	0.874	0.292	0.329	1.817	3.107	1.587	2.289	2.714	0.071
J5					1.000	3.271	2.884	1.357	0.292	0.415	2.154	2.884	0.550	2.154	3.271	0.077
J6						1.000	1.587	0.306	0.250	0.250	0.737	2.466	0.382	1.587	0.500	0.029
J7							1.000	0.415	0.347	0.361	0.303	2.621	0.630	1.063	0.754	0.030
J8								1.000	0.693	0.794	2.621	3.271	1.260	2.714	2.884	0.088
J9									1.000	1.260	4.000	4.327	4.000	4.000	3.979	0.156
J10										1.000	3.826	4.160	3.271	3.302	3.420	0.125
J11											1.000	2.520	0.342	1.587	0.737	0.038
J12												1.000	0.288	0.500	0.437	0.022
J13													1.000	3.302	2.714	0.078
J14														1.000	2.154	0.030
J15															1.000	0.037

We used Expert Choice, a decision support software that implements AHP, to obtain the priorities of user requirements. The priorities are shown in the last column of Table 4. We observe that J9: "Should keep daily, monthly, and yearly backup" and J10: "Security and confidentiality" have higher weights. The next higher weights are assigned to J8 and J3, which correspond to "Daily, monthly and yearly statistics" and "One and unique Medical Record Number (MRN) should be capable of giving the details of a particular patient", respectively. The prioritisation of user requirements is expected to be helpful to improve software quality by first addressing the factors that receive higher weights.

In order to identify technical requirements, we consulted several software engineers working in IIUM and IT unit of SQUH. The list is shown in Figure 1. As it is shown in the figure, all the technical requirements are divided into seven categories, namely, network, backup, programming language, security, application, user interface, integration and others.

Figure 1 List of technical requirements

1 2 3	Giga byte Ethernet network Broad band network Wireless technology	Network
4 5 6	Continuous backup (mirror) Write script to have automatic backup Use tape or other medium for backup	Backup
7 8 9	Use SQL Use 4GL Use Java	Programming language
10 11 12	Multi-level access Use of password and user name Smart card to authenticate	Security
13 14 15 16 17	Bi-lingual facility incorporation into application Initialise local language information Link to the database Application software Application access to the database	Application
18 19 20 21 22 23 24 25 26 27	Auto report generation Electronic forms that can accept various search requests from end-user Use general-purpose integrated package Process flow Choose and click options Online user manual/help Web-based application for easier access and easy management of application enhancement Internet facility to allow patient access through web Search engine to look at patient record by combination of names Use graphics and images	User interface
28 29 30	Incorporate Health Level 7 (HL7) and all international standard protocols Incorporate ORC and scanning facilities Use specific application packages	Integration and others

Figure 2 The QFD diagram for developing SQUH's quality healthcare software systems

Standard 9-3-1		
Strong	)	9
Moderate	*	3
Weak	6	1

			Network			Backup			Prog. Language			Security			Application	
			Giga byte Ethernet network	Broad band network	Wirelesstechnology	Continuous backup (mirror)	Write script to have automatic backup.	Use tape or other medium backup	Use SQL	Use 4GL	Use Java	Multi-level access	Use of password and user name	Smart card to authenticate	Bi-lingual facility incorporation into application	Initialise local language information	Link to the database
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
J1	Very fast	1	)	)	6				*	*	*						
J2	User friendly	2								)		*			*		
J3	One and unique MRN should be capable of giving the details of a particular patient	3				*				*							*
J4	Interface with other instruments	4		6	*												
J5	Automated lab reports and possibility to print all types of reports	5										*				6	6
J6	Departments stock check and update	6				*				*							)
J7	Incorporated internet fac ility	7		)		6											
J8	Daily, monthly and yearly statistics	8				*			*								)
J9	Should keep daily, monthly and yearly backup	9				)	)	)									
J10	Security and confidentiali ty	10				*						)	)	)			
J11	Scanning related patients' paper records and integrates them with Electronic Medical Record (EMR)	11															
J12	Capability for an advanced alert mail service	12															
J13	Very strong query system across all the system	13				6			)	*							*
J14	Should have facility to support financial information	14										*					
J15	Bi-lingual capabil ity (English/local language)….	15													*	)	
	Weight	1	0.63	0.97	0.28	2.49	1.40	1.40	1.17	1.35	0.21	1.63	1.12	1.12	0.29	0.41	1.62
	Columnnumber		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Ranking of the technical requirements		20	14	28	1	4	4	8	6	29	2	9	9	27	24	3

	Application		User interface										Integration and others				
	Application software	Application access to database	Auto report generation	Electronic forms that can accept various search requests from end-user	Use general-purpose integrated package	Process flow	Choose and click options	Online user manual/help	Web-based application for easier access and easy management of application enhancement	Internet facility to allow patient access through web	Search engine to look at patient record by combination of names	Use graphic and images	Incorporate Health Level 7 (HL7) and all international standard protocols	Incorporate ORC and scanning facilities	Use specific application packages	Importance of the WHATs	Row number
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
J1	*					*	6									0.07	1
J2				*	6	)	)	*	*	*	*	)				0.06	2
J3		*									*					0.08	3
J4	)				)								)	6	*	0.07	4
J5		*	)	*	*							*			*	0.07	5
J6	*															0.02	6
J7									*	*		6				0.03	7
J8			*													0.08	8
J9																0.15	9
J10															*	0.12	10
J11														)		0.03	11
J12									*	*						0.02	12
J13				)							*					0.07	13
J14					*										)	0.03	14
J15															*	0.03	15
Weight	0.93	0.49	0.98	1.11	1.02	0.76	0.62	0.18	0.34	0.34	0.68	0.81	0.63	0.41	1.2		
Columnnumber	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Ranking of the technical requirements	15	22	13	11	12	17	21	30	25	25	18	16	19	23	7		

6 Results

Having developed the technical requirements, we continued construction of the HOQ and established the relationships between the user and the technical requirements. As shown before, numerical values have been assigned to three different symbols in the relationship matrix. By using the formula (2) in Section 2, weights of all the technical requirements have been determined and these are shown in the third row from the bottom of the HOQ diagram shown in Figure 2. The diagram went through several revisions before reaching its present form. This was necessary in order to deliberate on the relationships among user requirements and technical requirements.

From the diagram, we observe that the continuous mirror backup ranks as the most important technical requirement. The second most important technical requirement that affects user satisfaction is the multi-level access. The third important technical requirement is the linkage to the databases. Table 5 shows the most important technical requirements belonging to each of the seven categories. Furthermore, we observe that all the technical requirements belonging to backup and security categories have appeared in the top 15 technical requirements, as shown in Table 6.

Table 5 The technical requirements of higher weights belonging to each category

No.	*Technical requirement*	*Category*	*Rank*	*Weights*
1	Continuous backup (mirror)	Backup	1	2.499
2	Multi-level access	Security	2	1.632
3	Link to the databases	Application	3	1.625
4	Use 4GL	Programming language	6	1.350
5	Use specific application packages	Integration and others	7	1.200
6	Electronic forms that can accept various search requests from end-user	User interface	11	1.119
7	Broad band network	Network	14	0.971

Backup capability is a desired characteristic of any information system. There are numerous instances where organisations have lost millions of dollars due to loss of data/information. It is always safer to have backup files for important data set. The matter has especially been felt at SQUH. As it is clear from the Figure 2, a number of user requirements will be satisfied by having continuous backup option in the system they use. In a hospital like SQUH, people from various departments need to access patients' records. Multi-level access of patients' records will greatly enhance user satisfaction. Linkage to the databases will enable the users to quickly retrieve the information they want. It is also directly related to a number of user requirements. In SQUH, only the authorised personnel should have the access to the information system. Higher weights of the items in security category necessitate proper and adequate authentication on the users' part before using the information system.

From Table 6, we also observe that at least one requirement belonging to each category has reached to the top 15 technical requirements. At the time of developing healthcare software for SQUH, the software engineers need to pay attention on the technical requirements that have received higher weights.

Table 6 Top 15 technical requirements

No.	*Technical requirement*	*Category*	*Rank*	*Weights*
1	Continuous backup (mirror)	Backup	1	2.499
2	Multi-level access	Security	2	1.632
3	Link to the databases	Application	3	1.625
4	Write script to have automatic backup	Backup	4	1.404
5	Use tape or other backup medium	Backup	4	1.404
6	Use 4GL	Programming language	6	1.350
7	Use specific application packages	Integration and others	7	1.200
8	Use SQL	Programming language	8	1.176
9	Use of password and user name	Security	9	1.125
10	Smart card to authenticate	Security	9	1.125
11	Electronic forms that can accept various search requests from end-user	User interface	11	1.119
12	Use general-purpose integrated package	User interface	12	1.022
13	Auto reports generation	User interface	13	0.986
14	Broad band network	Network	14	0.971
15	Application software	Application	15	0.936

7 Conclusion

Information System (IS) development projects are expensive and time consuming. Further, the failure rate of IS projects is higher compared to other kinds of projects. All care should be taken so that the systems developed satisfy users' needs. The present research has used the quality function deployment tool in helping the IT unit of SQUH to design a quality healthcare software system. Altogether, 95 staff from eight departments of SQUH participated in the study and articulated their needs in using information systems. A synthesised list has been prepared by considering the common and seemingly more important requirements from the lists generated by various focus groups. The list has been prioritised using AHP. Having obtained the prioritised user requirements, a list consisting of 30 technical requirements has been prepared in consultation with several software engineers. The technical requirements belong to seven categories, namely: network, backup, programming language, security, application, user interface, integration and others. At the end of the QFD exercise, we obtain a list of prioritised technical requirements. The requirements that have received higher weightages are: continuous mirror backup, multi-level access, linkages to the databases, *etc.* The findings of the research are expected to provide some guidelines to the engineers working in the IT unit of SQUH to develop the necessary hospital information system. It is also expected that in the design of the new system, if the technical requirements of higher priorities are taken into consideration, then the new system will enhance user satisfaction.

Since Hospital Information System is large, complex, and involves many different modules, we recommend that further studies be carried out to improve each module separately and then integrate them to form a single system. Further, for every department of the hospital, a separate HOQ diagram can be constructed to satisfy the department members' unique needs. The sample size may be increased and more people may be included from different hospitals and healthcare providers in order to make the findings generalisable.

Acknowledgement

The authors are grateful to the anonymous referees for their insightful comments which have been very useful in revising the paper.

References

Barnett, W.D. and Raja, M.K. (1995) 'Application of QFD to the software development process', *International Journal of Quality and Reliability Management*, Vol. 12, No. 6, pp.24–42.

Basak, I. and Saaty, T.L. (1993) 'Group decision making using the analytical hierarchy process', *Mathematical and Computers Modeling*, Vol. 17, pp.101–109.

Berk, J. and Berk, S. (2000) *Management for the Technology Sector*, MA, USA: Butterworth-Heinemann.

Bossert, J.L. (1991) *Quality Function Deployment: A Practitioner's Approach*, Milwaukee, Wisconsin: ASQC Quality Press.

Crawford, G.B. (1987) 'The geometric mean procedure for estimating the scale of a judgment matrix', *Mathematical Modeling*, Vol. 9, No. 3, pp.327–334.

Ermer, D.S. and Kniper, M.K. (1998) 'Delighting the customer: quality function deployment for quality service design', *Total Quality Management*, Vol. 9, Nos. 4–5, pp.S86–S91.

Ertay, T., Büyüközkan, G., Kahraman, C. and Ruan, D. (2005) 'Quality function deployment implementation based on analytic network process with linguistic data: an application in automotive industry', *Journal of Intelligent and Fuzzy Systems*, Vol. 16, pp.221–232.

Fitzpatrick, R. (2001) 'Strategic drivers of software quality: beyond external and internal software quality', *Proceedings of the Second Asia-Pacific Conference on Quality Software*, Hong Kong, pp.65–72.

Franceschini, F. and Marco, T. (1998) 'An application of quality function deployment to industrial training courses', *International Journal of Quality and Reliability Management*, Vol. 15, No. 7, pp.753–768.

González, M.E., Quesada, G., Picado, F. and Eckelman, C.A. (2004) 'Customer satisfaction using QFD: an e-banking case', *Managing Service Quality*, Vol. 14, No. 4, pp.317–330.

Grenci, R.T. (2004–2005) 'An adaptable customer decision support system for custom configurations', *Journal of Computer Information Systems*, pp.56–62.

Hagg, S., Raja, M.K. and Schkade, L.L. (1996) 'Quality function deployment usage in software development', *Communications of the ACM*, Vol. 39, No. 1, pp.42–49.

Hamilton, J. and Selen, W. (2004) 'Enabling real estate service chain management through personalized web interfacing using QFD', *International Journal of Operations and Production Management*, Vol. 24, No. 3, pp.270–288.

Hamza, A.R., Kwan, C.L. and Woods, P.C. (1999) 'Quality function deployment in construction design: application in low-cost housing design', *International Journal of Quality and Reliability Management*, Vol. 16, No. 6, pp.591–605.

Islam, R. (2003) *The Analytic Hierarchy Process: An Effective Multi Criteria Decision Making Tool*, Research Center, International Islamic University Malaysia.

Lin, W.T., Hung, Y.H., Huang, C.T. and Wu, C.C. (2005) 'A performance evaluation of the after-sales service information systems provided by the Taiwanese machine tool industry', *International Journal of Management*, Vol. 22, No. 1, pp.112–126.

Motwani, J. and Kathawala, Y. (1994) 'Implementing quality function deployment: a systems approach', *Total Quality Management*, Vol. 6, No. 6, pp.31–35.

Pai, W.C. (2002) 'A quality enhancing software function deployment model', *Information Systems Management*, pp.20–24.

Ross, M., Brebbia, C.A., Staples, G. and Stapleton, J. (1995) 'Software quality management', *Measuring and Maintaining Quality*, Glasgow, UK: Computational Mechanics Publications, Vol. 2.

Saaty, T.L. (1977) 'A sealing method for priorities in hierarchical structures', *Journal of Mathematical Psychology*, Vol. 15, pp.234–281.

Sanford, J.L. (2005) 'How useful is QFD?', *Quality Progress*, January, pp.51–59.

Silva, F.L.R.D. (2004) 'Combined application of QFD and VA tools in the product design process', *The International Journal of Quality and Reliability Management*, Vol. 21, Nos. 2–3, pp.231–252.

Tan, K.C., Xie, M. and Chia, E. (1998) 'Quality function deployment and its use in designing information technology systems', *International Journal of Quality and Reliability Management*, Vol. 15, No. 6, pp.634–645.

Terninko, J. (1997) *Step-By-Step QFD: Customer-Driven Product Design*, 2nd ed., Florida: St. Lucie Press.

Appendix: Information system user requirements

Department	*Requirements*
Anesthetic	• Once the patient selected from operation list, it should refresh all the modules on the anesthetic's desktop.
	• It should be possible to access the data related to the selected patient by navigating between modules.
Doctors	• Should be possible to access all required modules from the doctor's desktop.
	• Customisation (by adding or deleting modules to the desktop) based on the departmental administrative policy should be easy and flexible.
Nursing	• Scrollable list of Medical Record Number (MRN) and names of all the patients admitted in the ward.
	• Once the patient is selected from the ward list, it should refresh all the modules on the nursing desktop and it should be possible to access the data related to the selected patient by navigating between modules.
	• Customisation (adding or deleting modules to the desktop) based on the nursing administrative policy should be easy and flexible.
	• Give an option for printing out the list of patients and their ID number.
	• Grant an option of printing patient's list and medication due within specific hour at any time.
	• It should be possible to record the actual administration of drugs/treatments by clicking items in the displayed list.
	• Ward stock should be updated automatically.
	• Capability for an advanced 'Alert Mail' service.
Radiology	• Scrollable list of Medical Record Number (MRN) and names of all the patients admitted in the ward.
	• Once the patient is selected from the ward list, it should refresh all the modules on the nursing desktop and it should be possible to access the data related to the selected patient by navigating between modules.
	• Customisation (adding or deleting modules to the desktop) based on the nursing administrative policy should be easy and flexible.
	• Give an option for printing out the list of patients and their ID number.
	• Grant an option of printing patient's list and medication due within specific hour at any time.
	• It should be possible to record the actual administration of drugs/treatments by clicking items in the displayed list.
	• Ward stock should be updated automatically.
	• Capability for an advanced 'Alert Mail' service.

Appendix: Information system user requirements (continued)

Department	*Requirements*
Medical records	a Patient registration • Generation of a unique MRN for each new registration. • The unique MRN should identify the patient and all related activities and services he/she will receive during his/her lifetime. • It should have provision to capture a unique patient identification, for example I.C. or passport number. • Should support three distinct text fields for names and one for family name or tribe. • There should be a date of birth field with an automated mechanism to display the current age in years, months and days. • The age should be current whenever the record is accessed. • If the date of birth is unknown, the age should be captured and approximate date of birth calculated automatically. • Facility to merge patient records in the event of duplication of MRN or separated if erroneous merging. • Automatically link mother to newborn baby. • Ability to print label with basic patient demographic data. • Produce patient ID card with bar coded MRN either laminated format, printed-paper or embossed plastic card. • Restriction to specified patient clinical data should be possible. • Facility to classify the financial status of the patient. b Appointment scheduler • Maintain list of days the clinic is open; start, and end time. • List of consultants, medical officers, and the clinic where they are available. • Link the doctor's clinic time roster with their leave. • User-friendly display of available slots. • Fixing appointment for the selected patient by clicking over empty slots. • Possibility to display and print selected list of appointments physician wise, clinic wise or MRN wise. • Indication of type of appointment (routine, urgent or immediate). • Provision to print appointment slip in English and local language. c Electronic Medical Record (EMR) • Capability to display patient's clinical profile by episode. • Possibility to printout a list of appointments in any day for retrieval of patients' files. • Tracking of patients' file movement and electronically acknowledge the receipt of files and return them back. • It should be possible to have interactive display of MRN and demographic details of patients being registered in the outpatient clinics.

Appendix: Information system user requirements (continued)

Department	*Requirements*
	• Have an option to use barcode scanners for recording the movements of patient's files.
	• Flexibility for medical records director to grant or cancel various level of access right to the members of the departments.
	• Customisation by adding or deleting modules to the desktop, based on the departmental administrative policy, should be easy and flexible
	• Scanning related patients' paper records and integrating them with Electronic Medical Record (EMR).
Laboratory	• There should be an option to restrict the range of display to items to the specific user laboratory.
	• It should be possible to clear the display filter and review all items (investigation results, clinical information, treatment record, drug history) related to the selected patients.
	• General pathology laboratory should have provision for entering abbreviated SNOMED code against each entry.
	• System should interface bi-directional with blood analysers.
	• Customisation (by adding or deleting modules to the desktop) based on the departmental administrative policy should be easy and flexible
	• A billing system that depends on test and be attributed to clients.
	• Statistical analysis for the number of each test per month.
	• Interface with other instruments.
	• Capability for an advanced 'Alert Mail' service.
Pharmacy	• Assist clinical pharmacist in designing appropriate dosage requirements.
	• Perform various pharmacokinetic calculations.
	• Possibility to represent the data graphically.
	• Provide online pharmacokinetic monographs.
	• Integrated inpatient and outpatient systems so that the pharmacy will keep a single record.
	• Access level control for restricted drugs.
	• A comprehensive drug information system should be available.
	• Capability for an advanced 'Alert Mail' service.
Administration and Finance	• Very strong query system across all the functions.
	• Should have facility to support financial information such as account payable and receivable, profit and loss, patient costing, *etc.*
	• Bi-lingual capability (English/Local language) is required in specific areas such as patients' registration data and instruction to patients in the medication and investigation areas.
	• Capability to create training/test environment similar to the live system for demonstration and training.
	• Should have facility to support financial information.